

REMARKS/ARGUMENTS

This Response and the following remarks are intended to fully respond to the Office Action mailed November 28, 2005. In that Office Action, claims 1-26 were examined, and all claims were rejected. More particularly, claims 1-9 and 14-22 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Schloss et al. (USPN 6,249,844); claims 10, 11, 13, 23, 24 and 26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schloss in view of Mattson (USPN 5,434,992). Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, no claims are amended, canceled or added. Therefore, claims 1-26 remain present for examination.

Objection to Specification

The Examiner objected to the specification based on two informalities. The specification is being amended herein to correct the informalities. Applicants respectfully request the withdrawal of the objection.

Claim Rejections – 35 U.S.C. § 102

Claims 1-9 and 14-22 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Schloss et al. (USPN 6,249,844), hereinafter “Schloss.” Applicants respectfully traverse this rejection, because Schloss does not disclose all the elements of the claims.

Generally, embodiments of the present invention relate to methods of partial-page output caching of web pages. Upon receiving a client request for a web page, a file system is searched for an appropriate file for responding to the request. The file is retrieved and processed to create a page object that corresponds to the requested web page. As may be appreciated, the web page is often made up of one or more components. According to the present invention, if a component supports output caching, an output cache is searched to retrieve the cached *component*, which advantageously avoids the need to create the component. With the present invention, as long as some components of the requested web page are contained within the output cache, improved efficiency is realized by using the cached components. This method avoids the problems of prior art methods that cache entire

web pages, which only improve efficiency when the cache contains the exact web page (without any changes) that is being requested.

Turning now to the reference cited by the Examiner, Schloss is one such prior art reference that relates to the caching of an entire web page. *See Schloss*, col. 2, lines 58-61. Indeed, Schloss' goal is to improve the cacheability of the entire web page. In order to improve the cacheability of the entire web page, Schloss describes parsing a web page into segments, some of which are replaced with "fragment identities." *See Schloss*, col. 2, lines 58-61. In essence, the fragment identities are references or pointers to the actual fragments, which is the web page information being replaced. An important improvement of Schloss relates to the fact that the entire web page can remain "cached" even though the a portion of the web page may change as long as the fragment identity remains and points to that changed portion. Although interesting, given its dependence on caching the entire web page, Schloss does not and cannot disclose, either explicitly or inherently the method claimed in the present application, as discussed below.

For instance, claim 1 of the present application is directed to a method that operates on a server system that creates a page object having references to objects on the server computing system in response to the received request for information; and when the output cache contains a pre-rendered output data of the object referenced by the page object, retrieving the pre-rendered output data; inserting the retrieved pre-rendered output data into a hierarchical tree data model; rendering the components of hierarchical tree data model to create a rendered page; and sending contents of the created rendered page to the client computing system.

The server in Schloss never performs the noted steps claimed in claim 1. Schloss discloses that the server may process two possible requests from a client. The first relates to a request for an object, which is a request for a web page. *See Schloss*, col. 6, lines 16-21. Schloss processes this request by retrieving the web page (object) from a cache, if in the cache, or from another location if not in the cache. *See Schloss*, col. 6, lines 31-38. If the web page is retrieved from another location, it may be modified with fragments to make the web page more cacheable. *See Schloss*, col. 6, lines 31-38. The web page is then sent to the client, and optionally cached. *See Schloss*, col. 6, lines 41-43. During the processing of the web page request, the server never retrieves pre-rendered output data, much less insert pre-

rendered output data into a hierarchical tree data model, as required by claim 1. Indeed, the fragments that may be inserted into a web page are newly created fragments. As stated in Schloss, “in step 725, after receiving the object, the object parser (with details described with reference to FIG. 8) is invoked to analyze the object description and *create fragments.*” *Schloss*, col. 6, lines 38-41. In other words, Schloss never discloses any circumstances under which the fragments are retrieved, or are pre-rendered. Instead, the fragments are newly created, which is basically the opposite of what is claimed in the present application. Thus, the process of Schloss for handling a request for a web page is distinct from the method claimed in claim 1, which provides for the retrieval of pre-rendered output data when available, and insertion of the pre-rendered output data into a hierarchical tree data model.

Further, Schloss discloses that a server may also process client requests for fragments. *See Schloss*, col. 6, lines 16-21. This is applicable to the situation where the web page is cached on the client, and refers to fragments stored on a server, which the client requests when the web page is being rendered. *See Schloss*, col. 5, lines 7-9 and col. 9, lines 53-55. To process a request for a fragment, the server will determine what version of the fragment is being requested. *See Schloss*, col. 8, lines 5-8. The server will then retrieve the fragment from a fragment cache, or create the fragment being requested. *See Schloss*, col. 8, lines 9-20. The requested fragment is then returned to the requesting client. Schloss never discloses that in processing a client request for a fragment, that the server inserts pre-rendered output data into a hierarchical tree data model; renders the components of hierarchical tree data model to create a rendered page; and send the contents of the created rendered page to a client computing system. In processing a fragment request, the server merely retrieves or creates the fragment, and passes to the client. Thus, neither of the server processes disclosed by Schloss anticipate the method of claim 1.

In sum, Schloss does not anticipate claim 1, because it does not disclose a server process that performs *inter alia*, retrieving pre-rendered output data, when available; inserting the retrieved pre-rendered output data into a hierarchical tree data model; rendering the components of hierarchical tree data model to create a rendered page; and sending contents of the created rendered page to a client computing system.

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Manual of*

Patent Examining Procedures (MPEP) § 2131 (quoting Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)). Furthermore, “the identical invention must be shown in as complete detail as is contained in the ... claim.” MPEP § 2131 (quoting Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989)). Applicants respectfully submit that Schloss does not meet the legal requirements for anticipating claim 1 of the present application.

As the Examiner states, independent claim 14 has similar limitations as claim 1 and accordingly, claim 14 is allowable over Schloss for the same reasons as described above for claim 1. Moreover, claims 2-13 and 15-26 depend, directly or indirectly, from claims 1 and 14 respectively and are thus allowable for the same reasons.

Claim Rejections – 35 U.S.C. § 103

Claims 10, 11, 13, 23, 24 and 26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schloss in view of Mattson (USPN 5,434,992). The Applicants respectfully traverse this rejection.

Mattson discloses a method for dynamically allocating space in a partitionable cache. *See Mattson, col. 2, lines 42-44.* The method disclosed by Mattson includes the use of a least recently used (LRU) list of cached reference objects. *See Mattson, col. 2, lines 60-62.* Mattson also discloses the creation and maintenance of a data structure that dynamically approximates a range of subcache partition sizes. *See Mattson, col. 2, lines 62-64.* Finally, Mattson applies a heuristic for optimizing one or more partition distributions over the set of subcaches. *See Mattson, col. 2, lines 64-66.*

As stated above, Schloss does not disclose all the elements of independent claims 1 and 14, and Mattson does not make up for the deficiencies in Schloss. Thus, the Examiner has not established a *prima facie* case of obviousness with regard to claims 10, 11, 23, 24 and 26, which depend from claim 1 or claim 14, because all the elements of the claims are not found within the cited references.

Conclusion

This Response fully responds to the Office Action mailed November 28, 2005. Still, that Office Action may contain arguments and rejections that are not directly addressed by this Response because they are rendered moot in light of the preceding arguments in favor of

patentability. Hence, failure of this Amendment to directly address an argument raised in the Office Action should not be taken as an indication that the Applicant believes the argument has merit, or that the Applicant acquiesces to the argument. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

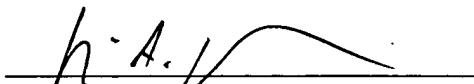
It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the foregoing remarks, it is believed that the application is in condition for allowance and thus prompt allowance is respectfully solicited. Since the remarks above are believed to distinguish over the applied reference, any remaining arguments supporting the claim rejections are not acquiesced to because they are not addressed herein.

Respectfully submitted,

Dated: February 28, 2006




René A. Pereyra, Reg. No. 45,800
Merchant & Gould P.C.
PO Box 2903
Minneapolis, MN 55402-0903
303.357.1639